

# 【jarvisoj刷题之旅】逆向题目[61dctf]stheasy的writeup

原创

iqiqiya 于 2018-09-09 16:57:34 发布 827 收藏 1

分类专栏: [我的逆向之路](#) [我的CTF之路](#) [我的CTF进阶之路](#) 文章标签: [逆向题目\[61dctf\]stheasy的writeup](#) [stheasy](#) [逆向题目\[61dctf\]stheasy的writeup reverse](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/xiangshangbashaonian/article/details/82559948>

版权



[我的逆向之路](#) 同时被 3 个专栏收录

108 篇文章 10 订阅

订阅专栏



[我的CTF之路](#)

92 篇文章 5 订阅

订阅专栏

[我的CTF进阶之路](#)

108 篇文章 18 订阅

订阅专栏

直接载入IDA

看有没有什么可疑字符串

Address	Length	Type	String
LOAD:080...	00000013	C	/lib/ld-linux.so.2
LOAD:080...	0000000F	C	libstdc++.so.6
LOAD:080...	0000000F	C	__gmon_start__
LOAD:080...	00000014	C	__Jv_RegisterClasses
LOAD:080...	00000015	C	__gxx_personality_v0
LOAD:080...	0000000A	C	libm.so.6
LOAD:080...	0000000E	C	libgcc_s.so.1
LOAD:080...	0000000A	C	libc.so.6
LOAD:080...	0000000F	C	_IO_stdin_used
LOAD:080...	00000007	C	fflush
LOAD:080...	00000005	C	puts
LOAD:080...	00000006	C	stdin
LOAD:080...	00000007	C	printf
LOAD:080...	00000007	C	strlen
LOAD:080...	00000007	C	memset
LOAD:080...	00000009	C	_IO_getc
LOAD:080...	00000012	C	__libc_start_main
LOAD:080...	0000000B	C	CXXABI_1.3
LOAD:080...	0000000A	C	GLIBC_2.0
.rodata:...	0000000C	C	Inmut flag:
.rodata:...	0000000F	C	Flag is right.
.rodata:...	0000000F	C	Flag is wrong.
.data:08...	00000034	C	k2j9Gh]AgfY4ds-a6QW1#k5ER_T[cvLbV7n0m3ZeX [CMt8SZo]U
.data:08...	00000005	C	3\$c!T
.data:08...	00000007	C	xxxx\x1B

<https://blog.csdn.net/xiangshangbashaonian>

双击进入

```

.rodata:080487E3          ad      0
.rodata:080487E4 ; char format[]
.rodata:080487E4 format      db 'Input flag:',0      ; DATA XREF: main+11f0
.rodata:080487F0 ; char aFlagIsRight[]
.rodata:080487F0 aFlagIsRight db 'Flag is right.',0      ; DATA XREF: main:loc_80486F8f0
.rodata:080487FF ; char s[]
.rodata:080487FF s          db 'Flag is wrong.',0      ; DATA XREF: main+39f0
.rodata:080487FF _rodata      ends
LOAD:0804880E ; =====
LOAD:0804880E
LOAD:0804880E
LOAD:0804880E ; Segment type: Pure code

```

<https://blog.csdn.net/xiangshangbashaonian>

双击进入引用

F5反汇编成c代码

经过分析可以看出sub\_8048630()这个方法是关键

双击进入

```

1 int __cdecl main()
2 {
3     char s; // [esp+10h] [ebp-110h]
4
5     printf("Input flag:");
6     sub_80485A0(&s, 0x100u);
7     if (sub_8048630(&s))
8         puts("Flag is right.");
9     else
10        puts("Flag is wrong.");
11    return 0;
12}

```

<https://blog.csdn.net/xiangshangbashaonian>

经过分析 再将一些难看的变量名什么的修改下

就变成下图这个样子

```

1 int __cdecl sub_8048630(char *s)
2 {
3     size_t v1; // eax
4     int i; // edx
5
6     if ( s )
7     {
8         v1 = strlen(s);
9         if ( v1 )
10        {
11            if ( v1 == 29 )
12            {
13                i = 0;
14                while ( s[i] == b[(a[i] / 3u - 2)] )
15                {
16                    if ( ++i == 29 )
17                        return 1;
18                }
19            }
20        }
21    }
22    return 0;
23}

```

<https://blog.csdn.net/xiangshangbashaonian>

分析可知s就相当于我们想要得到的flag 反过来我们可以通过已有的a和b这两个数组来得到flag

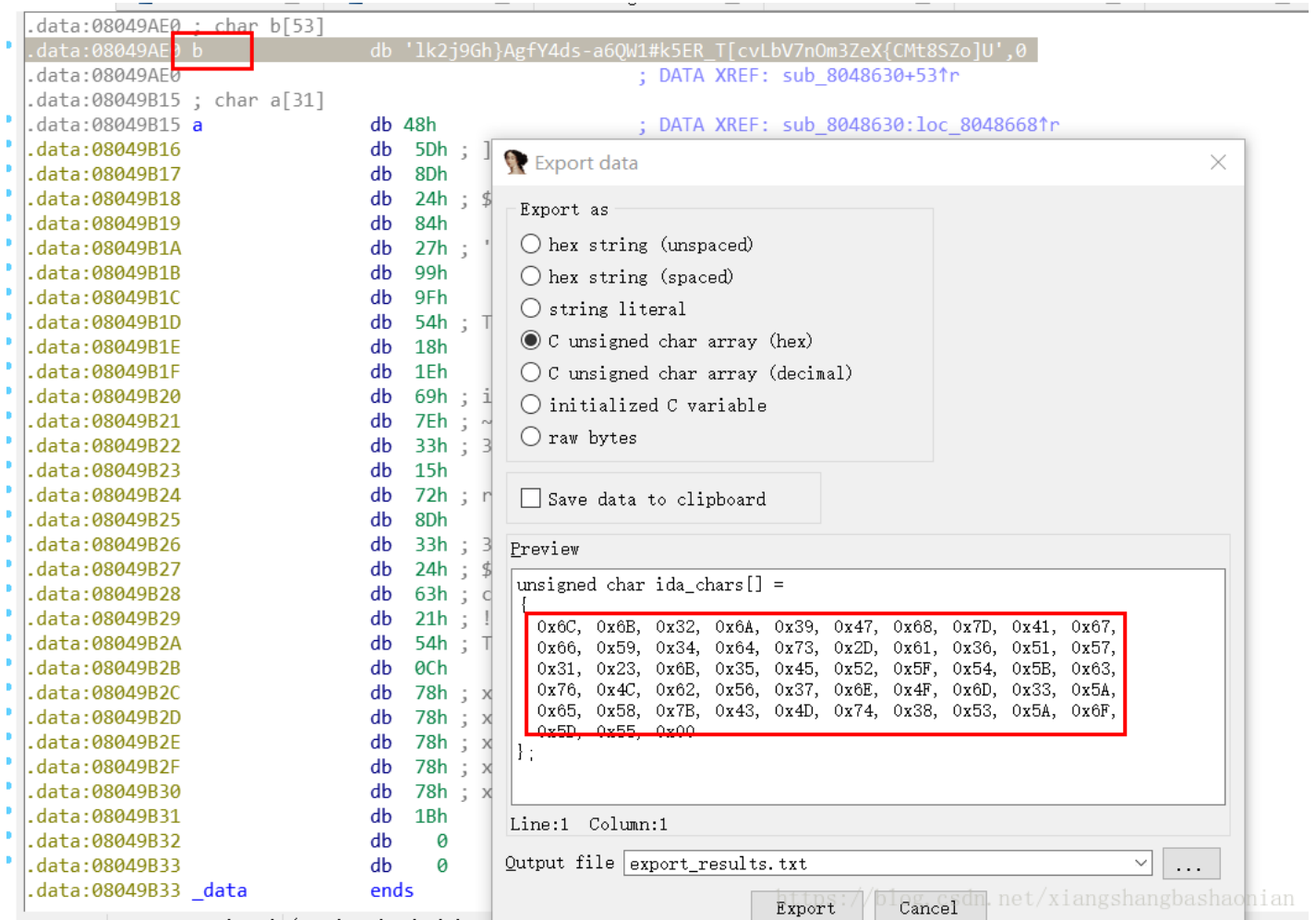
第一步：求 $a[i] / 3 - 2$  循环得到数组c(记得要将float转为int型)

第二步：求 $s[c[i]]$  循环即可得到flag

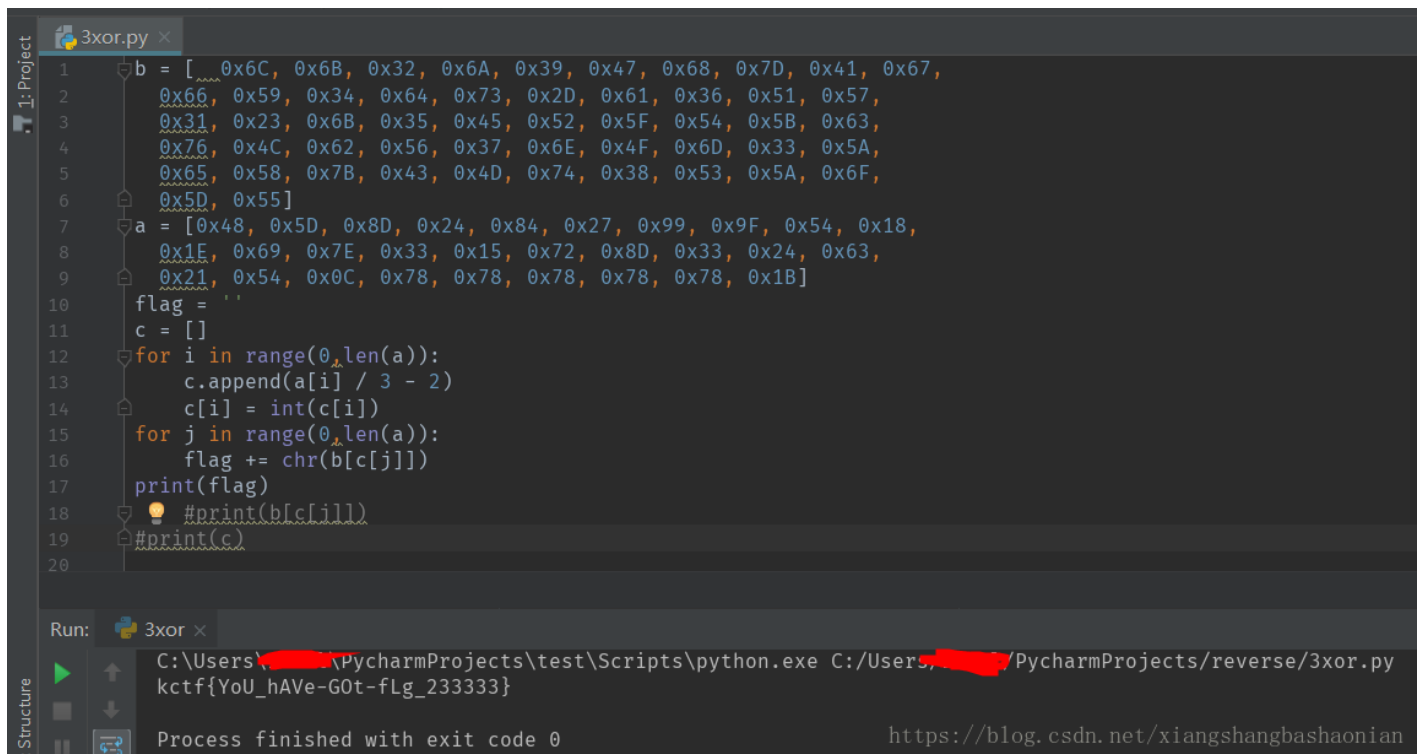
至于a,b的具体值可以双击进去找到

Shift+E就可以提取出来

The screenshot shows the IDA Pro interface. On the left, a list of memory addresses and their contents is displayed. A red box highlights the address `.data:08049B17` with the value `db 8Dh`. Another red box highlights a range of memory addresses from `.data:08049B17` to `.data:08049B31`, showing a sequence of `db` instructions with various hexadecimal values. The `Export data` dialog box is open on the right, showing the 'Export as' options. The 'C unsigned char array (hex)' option is selected. The 'Preview' section shows the resulting C code for the exported data, with a red box highlighting the hex values: `0x48, 0x5D, 0x8D, 0x24, 0x84, 0x27, 0x99, 0x9F, 0x54, 0x18, 0x1E, 0x69, 0x7E, 0x33, 0x15, 0x72, 0x8D, 0x33, 0x24, 0x63, 0x21, 0x54, 0x0C, 0x78, 0x78, 0x78, 0x78, 0x78, 0x1E`. The 'Output file' field is set to `export_results.txt`.



Py大法好:



最后提交成功

ctf2.b93676be23733b2fcd3988c1133c1c1

kctf{YoU\_hAVe-G0t-fLg\_233333}

SUBMIT

Correct Answer!!Congratulations!

<https://blog.csdn.net/xiangshangbashaonian>